УДК 004.032.26:581.522-047.44

НЕЙРОСЕТЕВАЯ СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ БАНКОВСКИХ РЕШЕНИЙ ПРИ ВЫДАЧЕ КРЕДИТОВ

Д.С. Сенькович
Магистрант кафедры информатики БГУИР, инженер-программист

А.В. Жвакина
Доцент кафедры информатики БГУИР, кандидат технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники, Республика Белапрусь
E-mail: dmitrysenkovich@gmail.com, zhvakina@bsuir.by

А.В. Жвакина
Образование: Белорусский государственный университет информатики и радиоэлектроники; кандидат технических наук, доцент. Научные интересы: нейронные сети, системы поддержки принятия решений, языки программирования, веб-программирование, бизнес-анализ, разработка требований к программному обеспечению, включая UML-моделирование, разработка приложений с графическим интерфейсом для решения прикладных задач. Более 80 публикаций, в том числе 14 учебных пособий.

Д.С. Сенькович
Образование: Белорусский государственный университет, Факультет Прикладной Математики и Информатики; Место работы: программист, "HES FinTech"; Научные интересы: нейронные сети, веб-программирование, облачные вычисления, распределенные системы, микросервисы.

Аннотация. Обсуждаются различные подходы к анализу информации о клиентах банка с целью принятия решения о выдаче кредита. Исследованы различные способы моделирования данной задачи, оценена их точность и время получения результатов.

Ключевые слова: анализ данных, нейронные сети, кредитование.

Залогом выгодного кредитования является оценка потенциального заемщика с точки зрения его платежеспособности и социальной благонадежности. Необходимо проанализировать возможность возврата кредита, опираясь на данные о предыдущей кредитной истории клиента, его платежеспособности и социальных особенностях.

Для решения данной задачи используются различные подходы. Один их них базируется на применении кредитного скоринга [1]. При этом рассматривается информация об имевшихся кредитах, и на основании статистической или математической обработки данных из базы кредитного регистра определяются:

– класс рейтинга;

– скорбалл;

– вероятность просрочки более 90 дней в год на определенную сумму.

Класс рейтинга присваивается на основании оценки в баллах (скорбаллов), характеризующей вероятность того, что кредит не будет возвращен, и вероятности просрочки выплат по кредиту. Чем ниже скорбалл, тем выше вероятность нарушения своевременного погашения кредита. Важными факторами являются время, прошедшее после заключения

первого договора о кредитной сделке, количество таких сделок, а также овердрафтовых и потребительских договоров, суммах и продолжительности просрочек.

Недостатком данного подхода является то, что его нельзя использовать по отношению к лицам, которые вообще не брали кредиты или брали их более 5 лет назад, заключали сделки о кредитах с лизинговыми или микрофинансовыми организациями.

Таким образом, чтобы принять решение о выдаче кредита, недостаточно опираться лишь на сведения о кредитном рейтинге, необходимо оценить также множество факторов, влияющих на возврат денег заемщиком: платежеспособность, которую характеризует уровень дохода и надежность компании-работодателя, демографические и социальные данные [2, 3].

В качестве параметров, используемых для принятия решения о выдаче кредита, выбраны следующие:

- размер кредита (amount);
- срок кредитования (term);
- сумма ежемесячного дохода (income);
- род занятий (occupation_type);
- наличие отчислений в пенсионный фонд(pension_contributions);
- судимость – наличие, отсутствие (criminal_record)
- семейный статус – женат, холост (marital_status)
- сумма пенсионого дохода(add_income_pension)
- звонок перед визитом(add_info_call_before_visit)
- мошенничество (add_info_fraud)
- недостаточность информации (add_info_inadequate)
- плохой внешний вид (add_info_poor_appearance)
- наличие задолженности (had_arrears)
- раньше были кредиты (had_credits_before)
- имеет активный кредит (has_active_credit)
- ежемесячный платеж (monthly_payment)
- официальный доход (pt_income)

Использовалось 1000 обезличенных наборов данных, описывающих реальных потенциальных заемщиков и принятое банком решение.

Так как исходные данные представлены разными типами, то предварительно необходимо было привести их в числовой вид и обработать: булевые значения (например, наличие пенсионных отчислений – да или нет) заменить на 1 и 0, данные, принимающие возможное значение из списка – на номер в данном списке; признаки, которые не содержат полезной информации, например, с одинаковыми значениями для всех клиентов отбросить.

Обрабатываемые данные случайным образом разделены на две группы: использованные для тренировки, и те, на которых проводилась независимая оценка качества, в соотношении 4:1, т.е. 800 и 200 записей. В свою очередь тренировочный набор данных разделен на выборки для обучения и валидации в соотношении 3:2.

Для прогнозирования вероятности погашения кредита использовались логистическая регрессия, метод опорных векторов (SVM), алгоритм случайного леса (*random forest*), нейронные сети.

Задача усложняется тем, что является несбалансированной – в наборе данных из 1000 клиентов только 90 получили отказ в выдаче кредита, поэтому при обучении подсчитывается максимальная оценка F1, выбирается модель с наилучшим ее значением, параметры, соответствующие данной модели, используются для оценки на тестовой выборке (*F1 score*).

В случае логистической регрессии исследовались алгоритмы оптимизации, представленные в таблице 1, использовано 100 различных случайных значений коэффициента регуляризации в интервале от 0 до 1.

Таблица 1. – Логическая регрессия

Алгоритм оптимизации	Вид регуляризации
liblinear	l1
sag	l1
newton-cg	l2
lbfgs	l2
sag	l2
saga	l2

SVM обучена с различными видами ядер и 100 различными случайными значениями коэффициента регуляризации в интервале от 0 до 1. Использованные ядра: *rbf, poly, sigmoid.* Линейное ядро не было использование, потому что этот способ использования SVM сводится к логистической регрессии.

Случайный лес обучался с использованием 100 случайных значений количества простейших алгоритмов (деревьев) в интервале от 50 до 500.

Недостатком вышерассмотренных подходов является то, что при аппроксимации нелинейных зависимостей количество параметров в таких моделях растет слишком быстро, что не характерно для нейронных сетей.

В экспериментах использовались полносвязная нейронная сеть (рисунок 1) и сеть для распознавания образов (*patternnet*), представленная на рисунке 3.

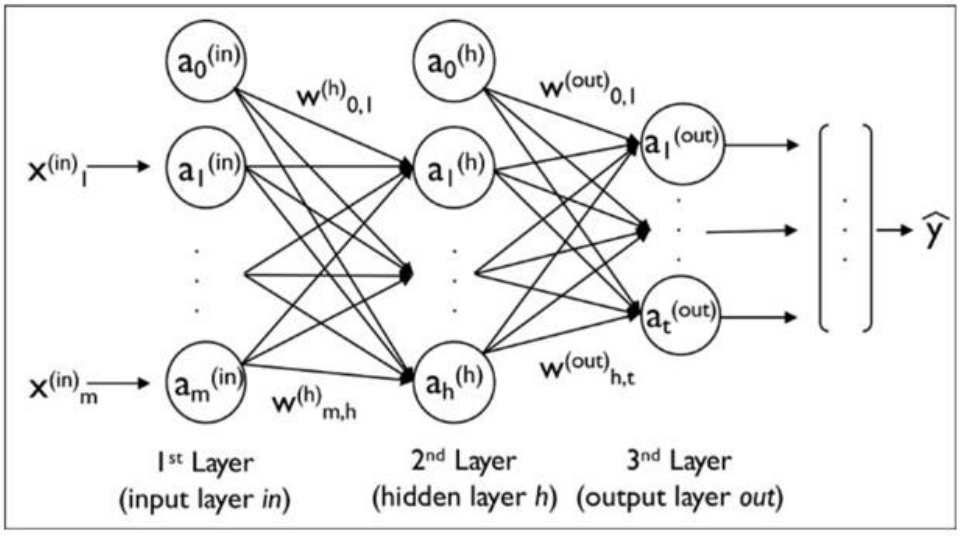

Рисунок *1.* – Полносвязная нейронная сеть

В экспериментах использованы модели с различным количеством внутренних слоев: от 1 до 7.

В качестве функции активации выбрана *ReLU*, которая имеет формулу *f(x) = max(0, x)* и представлена на рисунке 2.

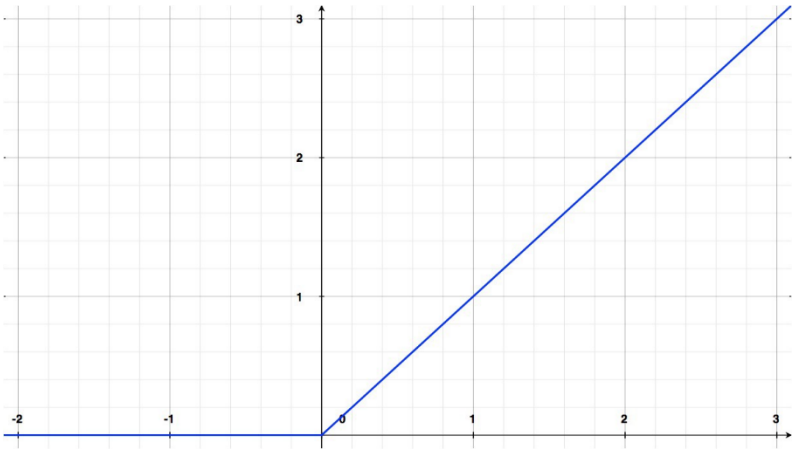

Рисунок *2*. – Функции активации

Достоинством данной функции по сравнению, например, с классической сигмоидой, является ее простота, так как это ускоряет вычисления [8]. Функция активации *ReLU* используется во всех экспериментах в полносвязных слоях.

Выходной слой модели имеет функцию активации *softmax*.

Инициализация весов нейронной сети может быть реализована простой выборкой небольших чисел, с небольшим разбросом около 0.01. Однако существуют исследования [5], в которых обосновывается выбор начальных весов из того или иного распределения в зависимости от количества нейронов на слое, функции активации и тому подобных параметров. Такие способы позволяют достичь лучших результатов на более ранних этапах обучения.

Примером может служить *glorot uniform* инициализатор. Это один из самый распространенных инициализаторов, использующихся для глубоких нейронных сетей [9, 10]. Особенно хорошие результаты в сходимости функция дает при использовании функции активации гиперболический тангенс. Вместо инициализации весов случайными числами из некоторого интервала данная схема инициализации выбирает значения из следующего интервала:

$$[-\sqrt{6 / (fan\_in + fan\_out)}; \sqrt{6 / (fan\_in + fan\_out)}],$$

где *fan_in* и *fan_out* длина векторов входных и выходных данных соответственно.

Математически доказано Хавьером Глоротом, что такая схема инициализации позволяет избежать возможной ситуации, при которой градиенты будут либо чересчур маленькими или вовсе нулевыми, что может привести к медленному обучения, а в худшем случае и отсутствии прогресса [7].

Другой довольно популярный выбор инициализатора - *He normal*. Этот способ выбирает значения из нормального распределения с математическим ожиданием 0 и разбросом $\sqrt{2 / fan\_in}$, где *fan_in* длина векторов входных данных. Этот способ лучше зарекомендовал себя с функцией активации ReLU, которая и используется в экспериментах.

Для каждой нейронной сети количество нейронов выбрано от 16 до 64 случайным образом. Каждый слой сети содержит именно это количество нейронов.

Каждая из нейронных сетей обучена в течение 50 эпох. Эпоха – изменение весов сети после прямого и обратного распространения на всех векторах выборки. Слишком маленькое значение ведет к более высокой вероятности получения более простой модели, неспособной выявлять сложные нелинейные зависимости. Слишком большие значения количества эпох ведет к возможному переобучению модели, проблемам *vanishing* и *exploding gradients*, когда

производные к ходе обратного распространения настолько малы, что обучение впадает в стагнацию и т.д. [4].

Коэффициент обучения задает скорость изменения параметров, ту часть градиента, которая используется для обновления весов. Любой алгоритм машинного обучения никогда не достигает абсолютного оптимума, а изменяется в некоторой его окрестности. При меньших значениях коэффициента алгоритм будет сходится медленнее, но в какой-то момент позволит приблизиться ближе к оптимуму функции потерь, в то время как большие значения коэффициента обучения позволяют быстрее обучать модель на первых порах, ухудшая результаты в будущем. Для каждой из нейронных сетей генерируется случайное значение коэффициента обучения из интервала от 0.1 до 0.001 по 10 на каждый из видов нейронных сетей.

Классическим алгоритмом обучения является *SGD* – стохастический градиентный спуск, оперирующий на подвыборках целой выборки для более быстрого прогресса в обучения [5, 6]. Возможным улучшением является использование метода *Momentum* – учет предыдущих значений параметров при очередном изменении. В этом помогает экспоненциально взвешенное скользящее среднее:

Экспериментально подтверждено, что стохастический градиентный спуск с *Momentum* в большинстве случаев лучше обычного[7].

RMSprop – еще один алгоритм обучения сети. Со временем параметры могут стать очень большими или очень маленькими, что приведет к очень большим и очень маленьким значениям градиентов соответственно (проблемы *gradient exploding* и *gradient vanishing* соответственно). *RMSprop* «выравнивает» значения параметров:

Приведенные выше алгоритмы обучения достаточно хорошо себя зарекомендовали на практике. Однако сейчас чаще используется еще один алгоритм – *Adam* [7]. Он сочетает в себе оба подхода, описанных выше, и в экспериментах показал наилучший результат.

Важным моментом для таких сетей является борьба с переобучением. Использовались два подхода:

– *L2*-регуляризация

– *dropout* регуляризация.

Смысл *L2*-регуляризации заключается в «штрафовании» модели за слишком большие веса, ограничивая таким образом значения каждого отдельного веса в сети:

$$D_{i,j}^{(l)} := \frac{1}{m}(\Delta_{i,j}^{(l)} + \lambda\Theta_{i,j}^{(l)}) \text{, if } j \neq 0$$

$$D_{i,j}^{(l)} := \frac{1}{m}\Delta_{i,j}^{(l)} \; if \; j = 0$$

Dropout регуляризация заключается в «выключении» некоторой части нейронов в слое. Как следствие, нейронная сеть не может «полагаться» на какие-то определенные нейронные, что результирует в регулиризационном эффекте.

В экспериментах использована *dropout* регуляризация. Для каждой нейронной сети генерируется *dropout* слой со случайным коэффициентом *dropout rate* – части «выключенных» нейронов – от 0.1 до 0.5. Этот коэффициент, по сути, вероятность «выключения» нейрона.

Исследовалась также нейронная сеть *patternnet*, которая при проверке на тестовых данных показала точность 96,7% (рисунок 4), общая точность – 92,4.

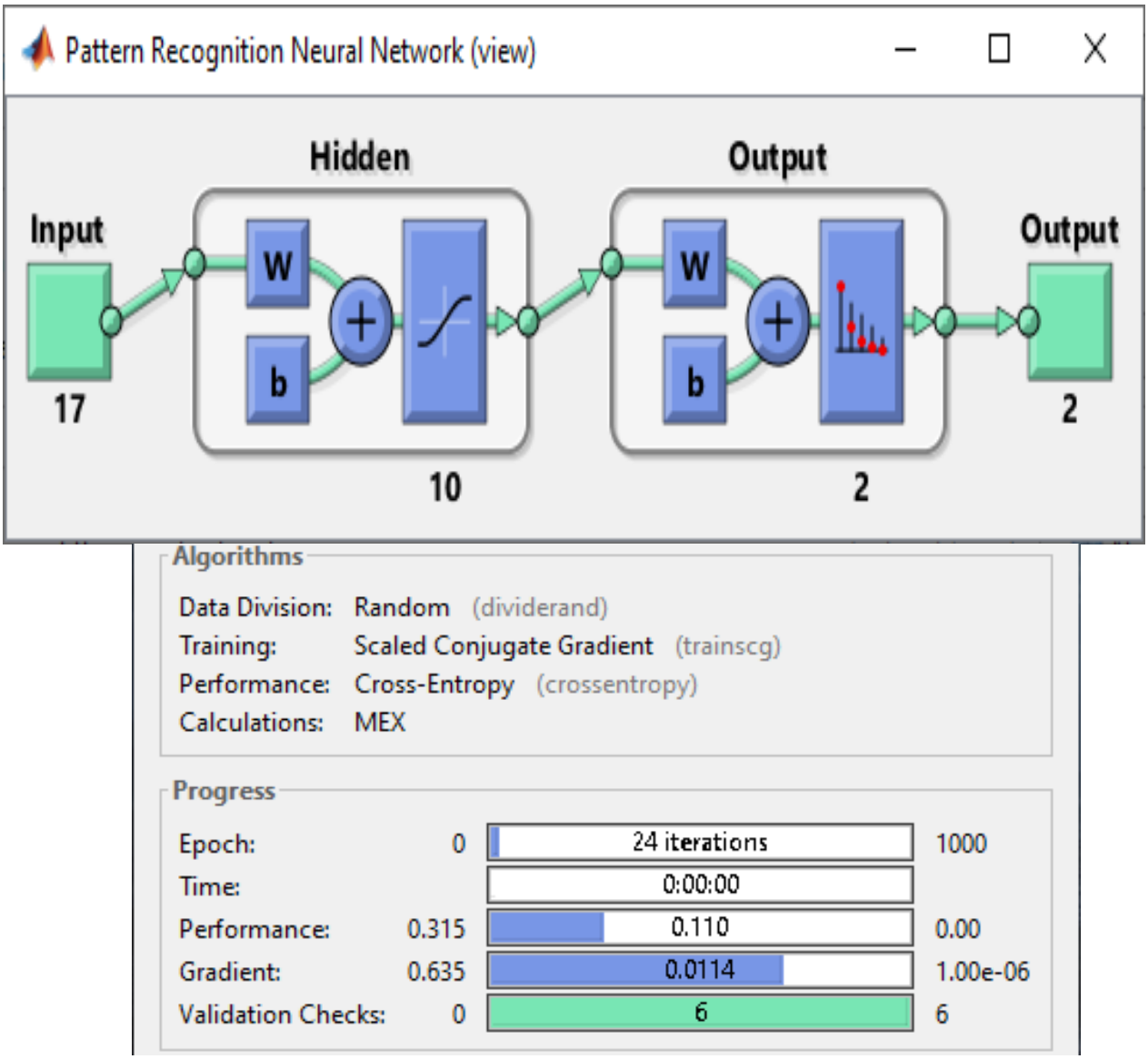

Рисунок *3.* – Сеть для распознавания образов

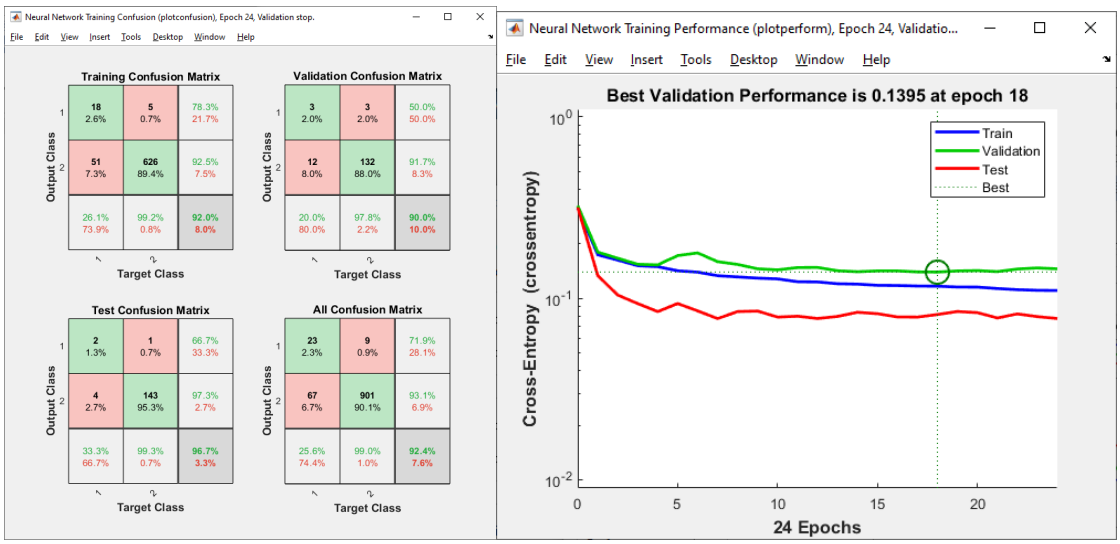

Рисунок 4. – Нейронная сеть *patternnet (1)*

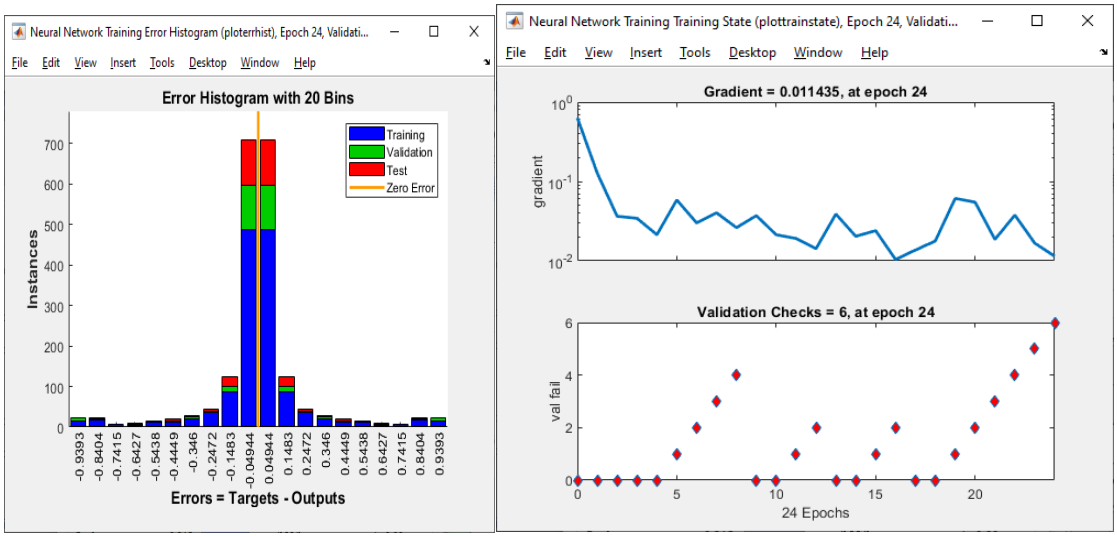

Рисунок 4. – Нейронная сеть *patternnet (2)*

Во втором слое использовалась функция softmax (рисунок 5)

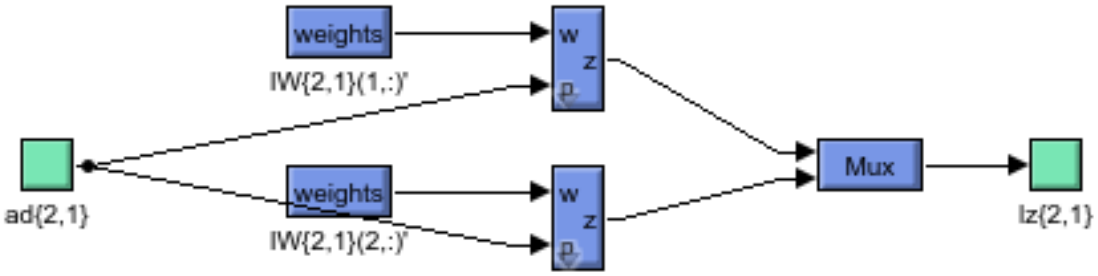

Рисунок 5. – Функция softmax

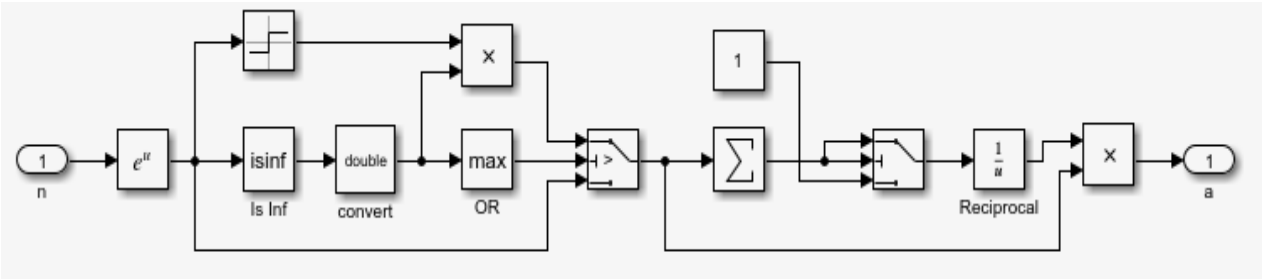

Рисунок 6. – Сеть

При экспериментах с сетью lvqnet (learning vector quantization neural network), представленной на рисунок 6, получено более длительное время обучения (8 сек.) и значения точности хуже (91%), что демонстрирует нецелесообразность использования данной сети для решения поставленной задачи.

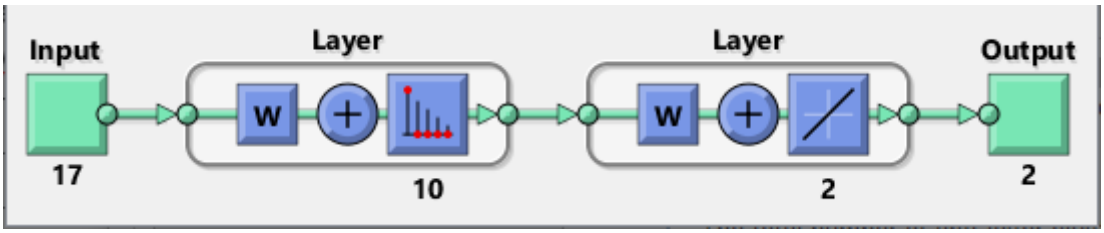

Рисунок 7. – Длительное время обучения

В таблице 2 представлены лучшие результаты обучения с лучшими параметрами у каждой из моделей, включая время обучения на все эксперименты для всех параметров.

Таблица 2. – Лучшие результаты обучения

Модель	Параметры	Время обучения	Время обучения лучшей модели	F1 оценка	Точность
Логистическая регрессия	Алгоритм оптимизации lbfgs, L2 регуляризация, C 0.23	40.94 секунд	0.0149 секунды	0.9595015576	90.5%
SVM	Ядро sigmoid, C 0.17	3.02 секунды	0.005 секунды	0.9565217391	91.1%
Случайный лес	265 базовых алгоритмов, построение деревьев до конца	32.95 секунды	0.34 секунды	0.9620253165	91.5%
Нейронная сеть	Оптимизатор adam, кол-во слоев 1, количество нейронов 61, dropout rate 0.4625, коэффициент обучения 0.0741, обучение в течение 50 эпох	23611.46 секунд ≈ 6.5 часов	4.925 секунды	0.9565217391	95.5%

Таким образом, в результате исследования различных подходов к поддержке принятия решения о выдаче кредита банку: логистической регрессии, метода опорных векторов (SVM), алгоритма случайного леса (random forest), нейронных сетей, набольшую точность показали нейронные сети. Использование нейронной сети позволит прогнозировать кредитоспособность потенциальных заемщиков и снизить кредитные риски банковских организаций.

Список литературы

[1]. Что такое кредитный скоринг? [Электронный ресурс]. – Режим доступа: https://creditregister.by/Help/WhatIsCreditScoring - Дата доступа: 16.02.2020

[2]. Как банки принимают решение о выдаче кредита [Электронный ресурс]. – Режим доступа: https://mycreditinfo.ru/kak_banki_prinimaut_reshenie_o_vydache_kredita - Дата доступа: 16.02.2020

[3]. Как банки проверяют заемщиков [Электронный ресурс]. – Режим доступа: https://www.sravni.ru/ enciklopediya/ info/kak- banki-proverjajut- zajomshhikov/ - Дата доступа: 16.02.2020

[4]. Simon Haykin. Neural Networks: A Comprehensive Foundation. (2nd Edition) / Simon Haykin. −Pearson Education, (South Asia :). 2006. − 823 p.

[5]. Laurene V. Fausett. Fundamentals of Neural Networks : Architectures, Algorithms and Applications. /Laurene V. Fausett. − Delhi, India: Pearson Education India, 2018. – 480 p.

[6]. Каллан Р. Нейронные сети : краткий справочник / Р. Каллан ; пер. с англ. и ред. А. Г. Сивака. – М.: Вильямс, 2018. - 279 с.

[7] Optimizers Explained - Adam, Momentum and Stochastic Gradient Descent / mlfromscratch.com [Электронный ресурс]. – Режим доступа: https://mlfromscratch.com/optimizers-explained/#/ – Дата доступа: 23.02.2020.

[8] What are the benefits of using ReLU over softplus as activation functions? / Quora [Электронный ресурс]. – Режим доступа: https://www.quora.com/What-are-the-benefits-of-using-ReLU-over-softplus-as-activation-functions – Дата доступа: 23.02.2020.

[9] Neural Networks and Deep Learning / Coursera [Электронный ресурс]. – Режим доступа: https://www.coursera.org/learn/neural-networks-deep-learning – Дата доступа: 23.02.2020.

[10] Deep Learning Specialization / Coursera [Электронный ресурс]. – Режим доступа: https://www.coursera.org/specializations/deep-learning – Дата доступа: 23.02.2020.

NEURAL NETWORK DECISION SUPPORT SYSTEM ABOUT LENDING

D.S. Senkovich
Master's Department
Informatics BSUIR, Software Engineer

A.V.Zhvakina
Associate Professor, Department of Informatics,
BSUIR, Candidate of Technical
Sciences, Associate Professor

Belarusian State University of Informatics and Radioelectronics
E-mail: dmitrysenkovich@gmail.com, zhvakina@bsuir.by

Absract. The possibilities of various approaches to analyzing information about bank customers with the aim of making a decision on granting a loan are discussed. Various methods of modeling this problem are investigated; their accuracy and time of obtaining results are estimated.

Keywords: data analysis, neural networks, lending.